



LAUREA
AMMATTIKORKEAKOULU
Yhdessä enemmän

QA-testaus osana ketterää mobiili- kehitystä

Nina-Anastasia Korpelainen

2017 Laurea



Laurea-ammattikorkeakoulu

QA-testaus osana ketterää mobiilikehitystä

Nina-Anastasia Korpelainen
Tietojenkäsittelyn koulutus
Opinnäytetyö
Marraskuu, 2017

Nina-Anastasia Korpelainen

QA-testaus osana ketterää mobiilikehitystä

Vuosi 2017

Sivumäärä 377

Opinnäytetyön tavoitteena oli tuottaa Yritys X:lle testaus suunnitelma ja testitapauksia mobiilisovellus Y:lle. Mobiilisovellus Y on erilaisten yritysten ja ryhmien käytössä oleva sovellus, jossa työnantaja luo työntekijöilleen tavoitteita ja tehtäviä, joihin työntekijät vastaavat.

Testaus suunnitelma sisältää mobiilisovellus Y:n testaukseen liittyvät tiedot. Testitapauksia on tehty kriittisistä kohdista, sisällöstä, syötteistä sekä mobiilista. Opinnäytetyössä otetaan huomioon laadunvarmistus, jotta sovellus olisi asiakkaan laatuvaatimuksien mukainen. Teoriaosassa perehdytään ketteriin menetelmiin, laadunvarmistukseen, testaukseen yleisellä tasolla sekä mobiilitestaukseen.

Testaus suunnitelman teossa käytettiin tietoja, jotka saatiin Yritys X:ltä liittyen mobiilisovellus Y:hyn. Tietoja saatiin myös mobiilisovellus Y:n kehittäjältä ja testaajalta. Testaus suunnitelma sisältää aikataulun ja muita tärkeitä asioita testaukseen liittyen. Testitapauksia on yhteensä 33 ja niitä on monia erilaisia.

Jatkossa testaus suunnitelmaan voidaan käyttää sovellus Y:n kanssa ja laajentaa myös muihin Yritys X:n sovelluksiin ja projekteihin. Testaus suunnitelmaa ja testitapauksia on hyvä uudistaa tietyin väliajoin, jotta uudet ominaisuudet ja päivitykset tulevat testattua kunnolla.

Asiasanat: Mobiilisovellukset, Testaus, Ketterä kehitys, Laadunvarmistus

Nina-Anastasia Korpelainen

QA-Testing as Part of Agile Mobile Development

Year	2017	Pages	377
------	------	-------	-----

The purpose of this Bachelor's thesis was to produce a testing plan and test cases for Application Y which belongs to Company X. The mobile application Y is used by different companies and groups and in it the employer creates tasks and goals for the employees which the employees answer.

The testing plan includes all the information about mobile application Y's testing. It includes test cases about critical parts, content, inputs and mobile. This thesis also ensures quality assurance so that the application would function according to the customers quality standards. The theoretical part consists about agile development, quality assurance, testing in general and mobile testing.

The testing plan was written based on information acquired from the Company X about Application Y. Information was also gotten by one of the developers working with Application Y and from the tester of Application Y. The testing plan contains a schedule and other important information about testing. There are 33 test cases in total varying in type.

In the future, the testing plan can be used with Application Y and with other applications and projects undergoing at Company X. It should be noted that the testing plan and test cases should be kept up to date at consistent intervals to ensure that new features and updates will be tested accordingly.

Keywords: Mobile applications, Testing, Agile Development, Quality Assurance

Sisällys

1	Johdanto	6
2	Työn lähtökohdat	7
2.1	Tutkimuskohteen kuvaus	7
2.2	Aihealueen rajaus	7
3	Laadunvarmistus	8
4	Ketterät menetelmät - Scrum	10
5	Testaus osana sovelluskehitystä	11
5.1	Testaussuunnitelman pääperiaatteet	13
5.2	Yleistä testitapauksien rakentamisesta	13
6	Tutkimusmenetelmät	14
7	Tutkimuksen toteutus	16
7.1	Testaussuunnitelman laatiminen	16
7.2	Testitapausten suunnittelu	18
7.3	Testausvälineet	18
8	Tutkimuksen tulokset	19
8.1	Testaussuunnitelma	19
8.2	Testitapaukset	22
9	Yhteenveto ja johtopäätökset	23
10	Jatkokehitysehdotukset ja oman oppimisen arviointi	24
	Lähteet	26
	Kuviot	28
	Taulukot	28
	Liitteet	29

1 Johdanto

Tämän työn aiheena oli testauksen kehittäminen ketterästi kehitetylle mobiilisovellukselle. Käytännön osuus sisältää testaussuunnitelman ja aitoja testitapauksia. Teoriaosuudessa käsitellään ketteriä menetelmiä yleisesti, scrumia, laadunvarmistusta sekä testausta. Tarkemmin syvennyttään erilaisiin testaustapoihin, joita Yritys X käyttää silloin tällöin ja joita tullaan käyttämään kunnolla Sovellus Y:n kanssa. Laadunvarmistus osuudessa puhutaan yleisesti laadusta ja sitten tarkemmin myös laadunvarmistuksesta. Opinnäytetyön tuloksena Yritys X sai toimivan testaustavan mobiilisovellukselle Y, johon kuuluu testaussuunnitelma ja oikeita testitapauksia. Testaussuunnitelma sisältää kaikki tärkeimmät asiat testauksen aikataulusta, henkilöistä ja metodeista.

Sovellus Y on Yritys X:n erittäin menestynyt ja käytetty mobiilisovellus, jonka testaus on jäänyt vajavaiseksi. Sovellus Y:tä ei ole missään vaiheessa testattu systemaattisesti, vaan sitä on testattu epäloogisesti aina silloin tällöin. Sovellus Y on tarkoitettu erityisesti erilaisille yrityksille ja ryhmille, jotka haluavat seurata työntekijöidensä tai ryhmän jäsenien edistymistä eri asioissa, niin kotona kuin töissäkin. Sovelluksessa työnantaja luo työntekijöilleen tavoitteita ja tehtäviä, joihin sitten vastataan määräaikaan mennessä. Sovelluksesta on olemassa myös tietokoneversio.

Yritys X käyttää kaikissa projekteissaan ketterää menetelmää, scrumia. Sitä käytetään myös sovellus Y:n kanssa työskennellessä ja testaus ajoitettiin sen mukaan. Testausta suorittaa tällä hetkellä yksi työntekijä, tasaisin väliajoin. Testauksen tavoitteena on varmistaa, että tuote toimii niinkuin pitääkin ja asiakkaan vaatimusten mukaisesti.

Testitapaukset luotiin Yritys X:lle jo aikaisemmin toisen henkilön avustuksella. Nyt niitä on käytetty joitakin kuukausia ja niistä on löytynyt omat hyvät ja huonot puolensa. Osa testeistä on automatisoitu ja Yritys X on ottanut myös käyttöön kuormitustestauksen. Testitapauksista on tarkoitus muokata vielä parempia ja toimivampia, jotta testaus sujuisi jouhevammin. Testitapauksia on siis muokattu testaajan ja muiden projektin kanssa työskentelevien ehdotusten ja ajatusten mukaisesti.

Tutkimus aloitettiin perehtymällä lähdekirjallisuuteen ja siten selvittäen, mitä on laadunvarmistus, ketterä kehitys ja testaus. Tietoja etsittiin myös internetistä ja käytettiin hyväksi muutaman eri henkilön haastattelua liittyen Sovellus Y:hyn ja testaukseen. Näiden pohjalta kehitettiin testaussuunnitelma ja testitapauksia sovellukselle Y. Ensimmäisistä testitapauksista saadun palautteen perusteella kehitettiin uudet testitapaukset, jotka olivat vanhoihin verrattuna huomattavasti parempia.

2 Työn lähtökohdat

Yritys X on tähän asti testannut mobiilisovelluksiaan vaihtelevasti ja usein vain testaten kriittisimmät kohdat. Käytössä on ollut niin sanottu Ad Hoc-testaus, jossa yksi henkilö testaa mielestään tärkeitä kohtia ja omaan ammattitaitoon perustuen päättelee, onko tuote valmis ja toimivatko tietyt ominaisuudet niin kuin pitääkin. Tämän opinnäytetyö tarkoituksena on tuottaa Yritys X:lle kattavampi ja johdonmukainen tapa toteuttaa testausta Mobiilisovelluksessa Y.

Työ lähti käyntiin Sovellus Y:hyn tutustumalla, jonka jälkeen sovellusta testailtiin ja siitä alettiin kehittää testitapauksia ja testaus suunnitelmaan. Opinnäytetyön johtopäätöksissä tuotiin esille testaus suunnitelma kokonaisuudessaan sekä muutama testitapaus. Liitteisiin lisättiin kaikki loput testitapaukset sekä testaus suunnitelma.

2.1 Tutkimuskohteen kuvaus

Tarkoituksena oli selvittää toimiva laadunvarmistusta käyttävä testausprosessi ketteriin mobiilikehitysprojekteihin. Tarkoituksena oli siis luoda testaus suunnitelma sekä testitapauksia Sovellukselle Y. Tämän pohjalta yritys voi lähteä laajentamaan testitapauksia myös muihin mobiilisovellus projekteihin.

Yritys X on Helsingissä toimiva pk-yritys, joka räätälöi erilaisia palvelukokonaisuuksia muille yrityksille ja asiakkaille. Yritys X:llä on töissä tällä hetkellä noin 20 henkilöä. Yritys X:n suosituin ohjelma on Sovellus Y, joka on käytössä monilla yrityksillä ja erilaisilla ryhmillä. Yritys X:llä on myös monia muita suosittuja ja toimivia ohjelmistoja sekä sovelluksia, joita käytetään paljon.

Sovellus Y on monien eri yritysten käytössä oleva sovellus, jolla työnantaja voi antaa työntekijöilleen erilaisia tehtäviä ja tavoitteita. Tällaisia tehtäviä voi olla muun muassa seuraavat: Mitä olet suoriutunut tämän viikon töistä?, Jaa kuva omasta työpäivästäsi tai Oletko muistanut ilmoittautua tulevaan seminaariin?. Näihin vastataan kuvalla, tekstillä tai jollakin muulla tavalla, minkä työnantaja on määritellyt. Tämän jälkeen työnantaja ja työntekijä voivat tarkastella muiden tehtäviä ja kommentoida niitä halutessaan.

2.2 Aihealueen raja

Tässä opinnäytetyössä keskitytään nimenomaan ketterästi kehitetyn mobiilisovelluksen testausprosessiin, johon kuuluu Yritys X:n kehittämä Sovellus Y. Sovellus Y:lle tehtiin myös testaus suunnitelma, joka keskittyy pelkästään mobiilipuoleen. Mobiilisovellus Y:lle kehitettiin myös testitapaukset, jotka ovat tarkoitettu mobiilisovelluksen testausta varten.

Tässä opinnäytetyössä ei käsitellä tietokoneversion testausta. Tässä opinnäytetyössä ei myöskään huomioida yritysten muita projekteja tai sovelluksia. Testaussuunnitelma ja testitapaukset koskevat Sovellus Y:n käyttäjäpuolta, Sovellus Y:n admin puoli on jätetty tämän tutkimuksen ulkopuolelle.

3 Laadunvarmistus

Sanalla laatu on monia erilaisia tulkintoja. Yleensä ihmiset mieltävät sanan laatu tarkoittavan huippulaatuista tai hyvää. Virheettömiä ohjelmia ja niiden suoritustehokkuutta sekä luotettavuutta pidetään yhtenä tapana mitata laadukkuutta. Tuotteen tai sovelluksen ominaisuudet ovat osa laatua, sillä yleensä tuote tai sovellus on hankittu tyydyttämään tiettyjä taseita. Ihmisten toiveet ja mielipiteet halutuista sovelluksen ominaisuuksista muuttuvat jatkuvasti, joten yrityksen tulisi kehittää sovelluksiaan koko ajan. (Haikala & Märijärvi 2004, 192-193)

ISO 9000:n mukaan laatu on aste, jolla joukko luontaisia piirteitä täyttää vaatimukset. Vaatimukset määrittelee yleensä asiakas tai tuotteella on jo omat laatuvaatimukset. Piirteet viittaavat tuotteen jyvyyteen ja asteella viitataan siihen, että laatu on jatkumo, se alkaa nolasta ja jatkuu eteenpäin. (Chemuturi 2011, 2-3.)

Laatu ja luotettavuus ovat punoutuneet toisiinsa ja ovat muutenkin erottamattomat. Laatu tarkoittaa, että tietty toiminallisuus toimii tiettyjen olosuhteiden aikana. Luotettavuus sen sijaan tarkoittaa, että tietyllä toiminallisuudella on tietty suorituskvyn taso, joka jatkuu läpi tuotteen elinkaaren. (Chemuturi 2011, 5-6.)

Laadunvarmistus tarkoittaa yksinkertaisimmillaan keinoja, joilla yritetään ottaa selvää jonkun tuotteen tai sovelluksen laadusta. Laadunvarmistus voi koskea koko yritystä, tiettyä yrityksen tuotetta tai vaikkapa tiettyä ohjelmistokehityksen vaihetta. Laadunvarmistuksen ideana on, että laadukas tuote saadaan laadukkaalla prosessilla. (Haikala & Märijärvi 2004, 267.)

Laadunvarmistus on suunniteltuja ja systemaattisia tapoja, joilla vakuutetaan johtoryhmä siitä, että määritellyt projektin standardit, käytännöt, toimintatavat ja menetit on käytössä (Chemuturi 2011, 9). Laadunvarmennus on työtä, jonka avulla tutkitaan tuotteen toimintoja ja niiden perusteella todetaan, että tuotteen laatumääritelmät täyttyvät (Kasurinen 2013, 108). Usein luullaan, että testaus on laadun tärkein työväline, mutta laadun tärkein lähde on suunnittelu- ja kehitystyö (Kasurinen 2013, 149).

Ohjelmistojen laadunvarmistus koostuu kahdesta eri osasta, ulkoisista ja sisäisistä ominaisuuksista ja niiden toiminnasta. Ulkoisen laadun määrittelee se, miten tuote toimii oikeissa olosuhteissa ja kuinka hyödyllinen se on käyttäjille, kun taas sisäisen laadun määrittelee taas itse koodi. Käyttäjät yleensä keskittyvät ulkoiseen laatuun ja käytettävyyteen. Ulkoinen laatu paranee sitä mukaa, mitä paremmaksi koodi kehitetään. (International Software Test Institute 2017.)

Tällä hetkellä on kaksi eri lähestymistapaa varmentaa ohjelmiston laatu, joko vikojen hallinta lähestymistapa tai laatuattribuuttien lähestymistapa. Kaikki asiat, jotka eivät ole linjassa asiakkaan vaatimusten kanssa, tulisi luokitella vioiksi. Usein kehittäjät eivät kokonaan ymmärrä asiakkaan vaatimuksia, joka johtaa lopulta suunnitteluvikoihin. Viat voivat johtua myös väärästä koodista tai huonosta tietojen käsittelystä. Vikojen hallinnassa on eri kategorioita, jotka määräytyvät vikojen vakavuuden mukaan. (International Software Test Institute 2017.)

Laatuattribuuttien lähestymistapa keskittyy kuuteen erilaiseen laatuajan tuntomerkkeihin. Toiminnallisuus tarkoittaa tärkeitä toimintoja, jotka ohjelmisto tarjoaa. Se sisältää viisi muuta alakohtaa, jotka muun muassa määrittelevät onko toiminnot toteutettu oikein, ovatko toiminnot sopivia, miten ohjelmisto toimii muiden systeemin komponenttejen kanssa, onko ohjelmisto linjassa tarvittavien lakien ja ohjenuorien kanssa ja pystyykö ohjelmisto käsittelemään dataa turvallisesti. Luotettavuus tarkoittaa sitä, miten ohjelmisto suoriutuu tehtävistään tietyissä oloissa ja ajassa. Se myös määrittelee sen, miten systeemi kestää komponenttien viat. Luotettavuuteen kuuluu kaksi alakategoriaa, jotka ovat ohjelman kypsyys ja palautuvuus. Ohjelman kypsyys tarkoittaa sitä, kuinka usein ohjelmistossa esiintyy vikoja. Palautuvuudella sen sijaan tarkoitetaan aikaa, jonka ohjelmisto tarvitsee virheistä toipumiseen. Käytettävyys kertoo kuinka helppoa ohjelmistoa on käyttää ja siihen kuuluvat ymmärrettävyys ja oppimiskyky. Ymmärrettävyys viittaa siihen, kuinka helppoa ohjelmistoa on ymmärtää ja oppimiskyvyllä viitataan siihen, kuinka paljon eritasoisten käyttäjien tulee panostaa ohjelmiston ymmärtämiseen. (International Software Test Institute 2017.)

Kolme seuraavaa kohtaa ovat tehokkuus, huollettavuus ja siirrettävyys, jotka kukin sisältävät monia alakohtia. Tiivistettynä tehokkuus riippuu hyvästä arkkitehtuurista ja koodauksesta. Huollettavuus riippuu suurimmalti osin siitä, kuinka monimutkainen koodi on ja siihen kuuluu vikojen pääsyiden tunnistaminen, kuinka helposti koodia voidaan muuttaa vian löytyessä, kuinka stabiili ohjelmisto on, kun sitä muutetaan ja kuinka paljon pitää panostaa ohjelmiston testaukseen. Viimeisenä kohtana on siirrettävyys, jolla tarkoitetaan ohjelmiston sopeutuvuutta ympäristön muutoksiin. Siihen kuuluu myös kuinka helposti ohjelmisto voidaan asentaa ja kuinka helposti komponentteja voidaan vaihtaa tietyssä ympäristössä. Seuraavan sivun kuvio 1:stä voidaan nähdä, mitä kaikki kohdat sisältävät. (International Software Test Institute 2017.)



Kuvio 1: Laatuattribuuttien lähestymistapa (International Software Test Institute 2017).

4 Ketterät menetelmät - Scrum

Ketterät menetelmät (Agile methods) on tapa kehittää ohjelmistoja ja sovelluksia ilman turhaa byrokratiaa ja nopeasti. Verrattuna vesiputousmalliin, ketterät menetelmät pyrkivät reagoimaan ongelmatilanteisiin nopeasti, kun taas vesiputousmallissa pyritään varautumaan kaikkiin mahdollisiin ongelmatilanteisiin. Ketterien menetelmien lähtökohtana voidaan pitää vuonna 2001 julkaistua ”Ketterän ohjelmistokehityksen julistusta”. (Kasurinen 2013, 25.)

Scrum on yksi ketterien menetelmien tapa, joka on yksinkertainen ja helppo toteuttaa. Monet yritykset käyttävätkin jo scrumia ja se sopii varsinkin yrityksille, jossa työntekijät tekevät usein yhteistyötä ja kommunikoivat vapaasti. Scrum-mallissa tehdään ensin vaatimusmäärittelyä ja suunnitelma, jonka pohjalta toteutus jaetaan yhden iteraation eli sprintin kokoiisiin pakketteihin, jotka sitten toteutetaan osa kerrallaan. Scrum-mallissa yksi sprintti voi kestää viikosta aina kuukauteen asti, jolloin toteutetaan valitut ominaisuudet, testataan ne ja hyväksytään. Sitten kun kaikki sprintit on suoritettu, tulisi tuotteen olla kaikin puolin valmis julkaisua varten. (Kasurinen 2013, 25-26.)

Scrumin kaikista tärkein osa on tiimin yhteistyö ja kommunikointi. Jokainen työpäivä aloitetaankin Scrum-palaverilla, jossa sovitaan kuka tekee mitään, mitä ongelmia on tullut vastaan ja mitä juuri sinä kyseisenä päivänä aikoo tehdä tuotteen eteen. Näiden kokousten avulla kaikki tuotteen kehittäjät ovat kartalla siitä, mitä kukakin tekee ja mitkä ovat kriittisimmät ongelmat juuri sillä hetkellä. (Kasurinen 2013, 26.)

Ensimmäinen scrum tiimi muodostettiin vuonna 1993, kun Jeff Sutherland halusi kokeilla uutta lähestymistapaa ohjelmistoprojektiin. Yhdessä Jeff McKennan ja John Scuminiotalesin kanssa he keksivät uuden metodin, scrumin. (Sims & Johnson 2011, 65-66.)

Ketterässä kehityksessä vaarana on, että vaatimukset kehittyvät ja ne täytyisi saada napattua (Farrel-Vinay 2008, 55). Vaatimusten kehittyessä ominaisuuksia pitää jatkuvasti kehittää ja muokata. Toisena vaarallisena kohta nähdään se, että sovelluksen kehittäjien täytyy olla erittäin motivoituneita, vastuullisia ja päteviä, jotta he dokumentoisivat kaiken mitä he tekevät (Farrel-Vinay 2008, 56). Yleensä kehittäjien täytyisi dokumentoida vaatimukset mahdollisimman pian. Jos mitään ei dokumentoida, on testitapausten tekeminen haastavaa ja tapausten tekeminen täytyisi aloittaa ilman syvempää tietämystä koko asiasta.

5 Testaus osana sovelluskehitystä

Testauksen pääperiaatteet on kirjoitettuna ISTQB-testaussyllabuksessa (ISTQB 2013), jossa on yhteensä seitsemän eri periaattetta. Nämä periaatteet ovat testaus osoittaa vikojen olemassaolon, täydellinen testaus on mahdotonta, aikainen testaus, vikojen kasaantuminen, hyönteismyrkkyparadoksi, testaus on tilanneriippuvaista sekä virheettömyyden harhaluulo. (Kasurinen 2013, 38-39.)

Testaus osoittaa, että vikoja on olemassa, mutta testaus ei voi todistaa, että vikoja ei ole ollenkaan. Vaikka koko sovellus testattaisiin kauttaaltaan, vikoja voi silti löytyä. Vikojen löytymättömyys ei tarkoita sitä, että sovellus olisi täydellinen. (ISTQB Exam Certification 2017) Monissa tapauksissa on tunnistettu, että vikoja on sovelluksessa, vaikka sitä olisi testattu erittäin tarkasti (Chavhan 2015).

Täydellinen testaus on mahdotonta, sillä kaikkia mahdollisia yhdistelmiä syötteitä ja alkuasetelmia on mahdoton testata. Sen sijaan pyritään hallitsemaan riskejä ja priorisoimaan, jotta testausta voidaan keskittää oikeille alueille. (ISTQB Exam Certification 2017.) Korkean riskin moduulit pystyy tunnistamaan, kun vaikutukset pystytään kartoittamaan (Chavhan 2015).

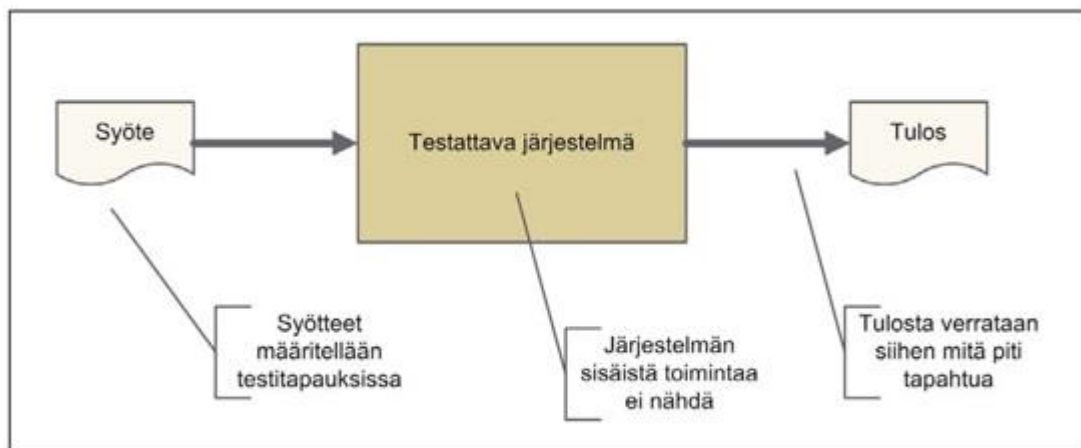
Testaus tulisi aina aloittaa mahdollisimman aikaisin, jotta bugit ja muut virheet huomataan jo alussa. Kun bugit huomataan heti alussa, niiden korjaaminen ei yleensä vie paljoa aikaa tai rahaa. Tämä auttaa julkaisemaan sovelluksen ajoissa ja odotetulla laadulla. (Chavhan 2015.) Usein viat kasaantuvat muutamiin sovelluksen osiin tai suurin osa toiminallisista vioista paikannetaan muutamista eri sovelluksen osista (ISTQB Exam Certification 2017).

Hyönteismyrkkyparadoksissa samoja testitapauksia käytetään kerta toisensa jälkeen ja lopulta ne eivät enää löydä uusia bugeja. On tärkeää tarkistaa testitapauksia tietyin väliajoin, jotta löydetään uusia bugeja ja vältetään hyönteismyrkkyparadoksi. (ISTQB Exam Certification 2017.) On sallittua lisätä uusia testitapauksia vielä testauksen jälkeenkin (Chavhan 2015).

Testaus on tilanneriippuvaista, koska erilaisia sivuja ja sovelluksia testataan eri tavoilla. Testauksen pitäisi olla erilaista ja olla riippuvainen testattavasta sovelluksesta. Tietokonesovel-

lusten ja mobiilisovellusten testausstrategiat ovat täysin erilaiset. (Chavhan 2015.) Jos soveluksen järjestelmä ei ole käytännöllinen ja se ei täytä käyttäjien tarpeita niin vikojen korjaaminen ei auta, tätä on virheettömyyden harhaluulo. (ISTQB Exam Certification 2017.)

Mustalaatikko-testaus (black box testing) tarkoittaa testaustapaa, jossa järjestelmälle syötetään erilaisia syötteitä ja sitten katsotaan miten se niihin reagoi. Tässä testaustavassa ei katsota mitä itse järjestelmä tekee vaan katsotaan konkreettisesti miten esimerkiksi sovellus reagoi väärään kirjautumisnimeen. Testitapauksiin kirjoitetaan valmiiksi erilaiset syötteet, jotka sitten testausvaiheessa testataan. Kaikista yleisin tapa on testata vaikkapa erilaisia syötteitä lomakkeeseen ja katsoa tuottaako sovellus virhekoodeja vai hyväksyykö se syötteet. Mustalaatikko-testauksessa kannattaakin kokeilla mahdollisimman monia eri syötteitä ja yhdistelmiä, jonka avulla saadaan helposti selville, miten järjestelmä reagoi yhdistelmiin, jotka eivät ole niin tavallisia. Vaikka järjestelmän sisäistä toimintaa ei tässä tavassa nähdäkään, on silti mahdollista todentaa tulokset oikeiksi katsomalla onko lopputulos toivotunlainen vai ei. Regressiotestit yleensä automatisoidaan, koska niitä tarvitsee toistaa aina uusien ominaisuuksien tai päivityksien jälkeen. Yritys X kuitenkin tekee ainakin toistaiseksi regressiotestauksen manuaalisesti. Kuten alla olevasta kuvio 2:sta näkee, testaamalla erilaisia syötteitä katsotaan, onko lopputulos oikea. Järjestelmän sisällä tapahtuvia asioita ei huomioida. (Kasurinen 2013, 51-52.)



Kuvio 2: Mustalaatikko -testaus (Kasurinen 2013, 52).

Regressiotestaus ei varsinaisesti ole oma testaustapansa, vaan sen on testauksen ominaisuus. Regressiotestaus on yleistä silloin, kun järjestelmään on tullut uusi ominaisuus tai päivitys, jonka takia järjestelmä tulisi testata mahdollisten virheiden varalta. Regressiotestauksen tärkein tehtävä on varmistaa, että uudet ominaisuudet tai päivitykset eivät ole rikkoneet jo aiemmin korjattuja ongelmia. (Kasurinen 2013, 54.)

Usein regressiotestauksessa vain koko ohjelmiston kattava testaus tuo kaikki mahdolliset virhetilanteet esille. Tähän kuitenkin ei yleensä ole tarpeeksi aikaa tai rahaa, jolloin testitapaukset pitäisi suunnitella niin, että mahdollisimman paljon virhetilanteita saataisiin selville ilman, että pitää testata koko järjestelmä uudelleen. Regressiotestausta suoritettaessa on muutenkin mietittävä järkevää laajuutta, mitä testataan ja mitä ei. Yleensä uuden päivityksen tai ominaisuuden jälkeen tulisi testata kaikki asiat, joihin nämä uudet ominaisuudet vaikuttavat ja asiat, joissa on aikaisemmin ollut ongelmia. (Tuovinen 2013, 17-18.)

5.1 Testaussuunnitelman pääperiaatteet

Testaussuunnitelma on dokumentti, jossa kerrotaan yksityiskohtaisesti siitä, mitä testataan, miten ja milloin. Siinä kerrotaan vähintään testauksen päälinjaukset. ISO/IEC 29119-testausstandardissa on määritelty tiettyjä perusasioita, jotka tulisi sisällyttää testaussuunnitelmaan. Näitä ovat muun muassa projektin perustiedot, testattavan tuotteen kuvaus, testausstrategia, kuinka laajoja testien tulisi olla, aikataulu ja toimintasuunnitelma. Pienemmissä yrityksissä voidaan tosin tehdä vielä yksinkertaisempi testaussuunnitelma, johon ei kuulu esimerkiksi testausstrategiaa. Pienemmissä yrityksissä voidaan käyttää vaikka SPACE DIRT-menetelmää, joka on lyhenne sanoista scope, people, approach, criteria, environment, deliverables, incidentals, risks ja tasks. (Kasurinen 2013, 90-91.)

Testaussuunnitelman tavoitteena on tyypillisesti kertoa yksityiskohtaisesti kaikista toimista, joita tarvitaan testauksen suorittamiseen tehokkaasti. Tärkeää on myös kertoa testausvälineistä sekä testiympäristöistä, jotta testit voidaan suorittaa oikealla ja mahdollisimman totuudenmukaisella tavalla. Koko testaussuunnitelman toimiminen viime kädessä riippuu kuitenkin aikataulusta ja siitä, kuinka hyvin testaussuunnitelmaa noudatetaan. (Farrel-Vinay 2008, 133.)

5.2 Yleistä testitapauksien rakentamisesta

Testitapaukset ovat erillisiä tapauksia, joissa kerrotaan askel askeleelta mitä testaajan tulee tehdä ja minkälaisen vastauksen hänen tulisi mistäkin kohdasta saada. Testitapauksessa määritellään myös tarkasti käytettävät versiot, laitteet ja muut testaamiseen tarvittavat välineet. Testitapauksia lisätään aina sitä mukaan, kun sovellukseen tulee uusia ominaisuuksia tai päivityksiä. On olemassa muutama hyvä ohje, jonka perusteella voidaan määritellä, mistä testitapauksia kannattaa kehittää. Uusia testitapauksia kannattaa suunnitella aina kun sovellukseen tulee uutta koodia, uusi päivitys, uusia ominaisuuksia, jos koodia on muutettu tai korjailtu ja silloin, kun sovellusta pitää testata uudella laitteella. (Kasurinen 2013, 92-93.)

Yksinkertaisimmillaan testitapauksiin kirjataan mitä testaajan tulee tehdä ja miten järjestelmän pitää siihen reagoida. Usein kuitenkin testaukselle pitää asettaa ehtoja, kuten missä tilassa järjestelmän pitää olla tai millä laitteilla pitää testata. Usein testitapauksiin liitetään

myös vaatimuksia ja muita ohjeita testaajalle. Näihin ohjeisiin kuuluu muun muassa kuka testitapaukset on suunnitellut, keneltä saa lisätietoja tapauksesta ja miten testitapaus vaikuttaa järjestelmään. (Kasurinen 2013, 93.)

Mobiililaitteilla tehdään kahdenlaista testausta, sovellustestausta tai laitteistotestausta. Laitteistotestauksessa testataan laitteiden toimivuutta ja sovellustestauksessa testataan mobiilisovellusten toimivuutta mobiililaitteissa. Mobiilisovelluksia testatessa pitää miettiä, että soveluksia on erilaisia, mobiilinettisovelluksia, paikallisia laitteessa olevia sovelluksia sekä niiden yhdistelmiä. (Ratnakar 2017.)

Sovellusten testaaminen mobiililaitteilla on hankalampaa kuin selaimessa testaaminen, sillä mobiililaitteita paljon erilaisia. Mobiililaitteita on eri kokoisia, mallisia, erilaisilla mobiilikäyttöjärjestelmillä ja eri käyttöjärjestelmäversioilla. On mahdoton testata näitä kaikkia, joten usein keskitytään kaikista suosituimpiin ja yleisimpiin laitteisiin sekä käyttöjärjestelmiin. (Ratnakar 2017.)

Muutamit asiat erottavat mobiilisovellusten ja tietokonesovellusten testauksen toisistaan. Tietokoneella sovellukset testataan prosessorilla ja mobiilissa sovellukset testataan matkapuhelimilla, kuten Samsungilla, Nokialla tai Applella. Mobiililaitteiden näyttöjen koot ovat huomattavasti pienemmät kuin tietokoneilla ja mobiililaitteissa on vähemmän muistia kuin tietokoneessa. Mobiililaitteilla käytetään yleensä langattomia yhteyksiä tai muita internet-yhteyksiä, kuten 2G, 3G tai 4G. Tietokoneella sen sijaan käytetään yleensä laajakaistayhteyttä. Mobiilisovelluksen testauksessa pitää ottaa huomioon myös akun kesto, sovelluksen nopeus, muistivaatimukset ja sovelluksen toiminnallisuus. (Ratnakar 2017.)

6 Tutkimusmenetelmät

Tutkimusmenetelmänä tässä työssä käytetään toiminallista kehittämistyötä, johon yhdistyy teorian esittäminen, joka kerätään käyttäen laadullista eli kvalitatiivista tutkimusta. Tiedonkeruumenetelminä tässä työssä käytettiin haastattelua, havainnointia sekä dokumentteja. Haastattelu on tiedonkeruutapana joustava, koska haastattelun aikana voidaan oikaista väärinkäsityksiä tai esittää tarkentavia kysymyksiä. Haastattelussa tärkeintä on saada tietoa halutusta asiasta niin paljon kuin mahdollista. (Tuomi & Sarajärvi 2009, 73.)

Kaikista keskeisimmässä osassa tässä työssä tiedonkeruumenetelmistä on havainnointi. Havainnoinnin avulla saatiin paljon tärkeää tietoa testitapausten toimivuudesta ja soveltuvuudesta. Osallistuvassa havainnoinnissa tutkija toimii muiden tiedonantajien kanssa yhteistyössä. (Tuomi & Sarajärvi 2009, 81.) Tätä opinnäytetyötä tehdessä toimittiin myös osana yritystä, joka antoi uudenlaista näkökulmaa testitapausten ja testausuunnitelman kehittämiseen. Oma-kohtaisen kokemuksen avulla uusia testitapauksia oli helppo luoda.

Toiminnallinen opinnäytetyö koostuu kahdesta eri osasta, toiminnallisesta osuudesta sekä opinnäytetyöraportista. Toiminnallisessa osuudessa kehitetään jokin konkreettinen tuote, palvelu, tapahtuma tai asia ja opinnäytetyöraportti koostuu teoriasta. Toiminnallisessa opinnäytetyössä ei riitä pelkän kirjan tai tapahtuman teko, vaan mukana pitää olla myös teoreettista tietoa. Tarkoituksena on yhdistää opitut asiat käytännön tuotokseksi. Usein teoriapohjaksi riittää jonkin aiheeseen liittyvän käsitteen tai käsitteiden perusteellinen selittäminen. (Vilkkä & Airaksinen 2004, 41-43.) Toiminnallisessa tutkimuksessa luotettavuus perustuu käyttäyhin aineistonkeruumenetelmiin sekä analyysiin.

Laadullinen tutkimus tarkoittaa tutkimusta, joka toteutetaan ilman tilastollisia menetelmiä. Määrällinen tutkimus perustuu lukuihin, kun taas laadullisessa tutkimuksessa keskitytään sanoihin ja lauseisiin. Määrällisessä tutkimuksessa kerätään ensin aineistoa, joka sitten myöhemmin analysoidaan ja sen pohjalta kirjoitetaan tutkimuksen tuloksia. Laadullisessa tutkimuksessa analysointi on mukana kaikissa tutkimuksen vaiheissa, sillä kerättävää tietoa on paljon. Aineistoa kerätään niin kauan kunnes ilmiöstä tai muusta tutkittavasta asiasta on saatu selville tarpeeksi tietoa ja tutkija ymmärtää mistä ilmiössä on kyse. (Kananen 2017, 35.)

Laadullisessa tutkimuksessa asioita tarkastellaan kokonaisuutena, kun taas määrällisessä tutkimuksessa keskitytään tilastoihin ja määriin. Laadullisessa tutkimuksessa vastaukseksi eivät kelpaa todennäköisyydet, vaan asiat pitää selvittää pohjia myöten, jotta voidaan olla varmoja tietojen oikeellisuudesta. (Alasuutari 2011, 38.)

Laadullisessa tutkimuksessa tutkittavan ilmiön tai asian teoriaa ei entuudestaan tunneta, mutta määrällisessä tutkimuksessa teorian tunteminen ennen tutkimuksen tekoa on välttämätöntä. Laadullisessa tutkimuksessa yritetään löytää vastaus kysymykseen ”Mistä on kyse?”. Jos aineistoinkeruvaiheessa pystyy keksimään tarkkoja kysymyksiä, olisi kyseessä määrällinen eikä laadullinen tutkimus. Laadullisen tutkimuksen ideana on saada syvälinen käsitys tutkittavasta aiheesta. (Kananen 2017, 32-33.)

Testitapauksien viitekehyksenä toimii internetistä löytyneet ohjeet ja mallit sekä painettu kirjallisuus. Kirjallisuudesta haetaan jonkilaista tietoperustaa sille, miten testitapaukset tulisi rakentaa, jotta validiteettia ja reabiliateettia löytyisi tuotoksista. Testitapaukset pohjautuvat kahden henkilön haastatteluun ja omaan kokemukseen, joiden pohjalta testitapauksia ja testaussuunnitelma luotiin. Jos pystytään osoittamaan, että saatu tieto vastaa teoriaa, niin silloin tulos on validi (Virtuaaliammattikorkeakoulu 2007). Hyvään validiteettiin päästään, kun tutkimuksen tulokset vastaavat tutkimukselle asetettuja tavoitteita (Heinonen 2012). Reliabiliteetillä tarkoitetaan sitä, onko mahdollista toistaa samoja tuloksia. Aineistoa analysoidessa voidaan vertailla samalla hankitavalla saatua tietoa keskenään (Heinonen 2012).

7 Tutkimuksen toteutus

Tutkimus toteutettiin yhdessä Yritys X:n kanssa. Aluksi piti tehdä vain testitapaukset, mutta myöhemmin päätettiin liittää myös testaussuunnitelma mukaan, jotta kaikki testaamiseen liittyvät asiat saataisiin koottua yhteen dokumenttiin. Testausdokumentti on Word-tiedosto, joka myöhemmin toimitetaan Yritys X:n omaan Jiraan. Jira on vikojen ja projektien seuranta ohjelmisto, jossa yritys voi ilmoittaa löytämistään bugeista, virheistä ja projekteihin liittyvistä asioista muille työntekijöille.

Tutkimuksessa käytettiin apuna lähdekirjallisuutta, artikkeleita sekä muutaman henkilön haastattelua. Näiden pohjalta luotiin testitapauksia ja testaussuunnitelma, jonka avulla Yritys X voi testata Sovellus Y:tä johdonmukaisesti. Erilaisia malleja testitapauksista etsittiin internetistä ja niiden sekä haastattelujen pohjalta luotiin toimiva testitapauspohja, johon kaikki testitapaukset rakennettiin. Osittain testitapausten teko perustui myös omiin kokemuksiin testitapauksia testatessa, sillä testauksessa oltiin mukana yrityksen sisällä.

7.1 Testaussuunnitelman laatiminen

Yritykselle X tehtiin mobiilisovellukselle Y testaussuunnitelma, jossa päädyttiin käyttämään SPACE DIRT-menetelmää, joka sopii paremmin pienemmälle yritykselle. SPACE DIRT-menetelmä sisältää eri osa-alueita, joita seuraavaksi tarkennetaan. Testaussuunnitelma alkaa Scope eli laajuuden määrittelyllä. Siinä määritellään mitä testataan ja mitä ei. Seuraavana on People eli ihmiset, jossa kerrotaan kuka tekee mitäkin, minkälainen koulutus testaajilla tulisi olla taustalla ja minkälainen aikataulu testauksella on. Tämä jatkuu osalla approach eli lähestymistapa, jossa kerrotaan, mitä testausmenetelmiä tullaan käyttämään. Tämän jälkeen tulee criteria eli kriteerit, joka sisältää kriteerit siitä, milloin aloitetaan, keskeytetään, jatketaan tai lopetetaan testaus. Seuraavaksi on environment eli ympäristö, jossa selvitetään minkälainen ympäristö testausta varten tulisi olla. Tätä jatketaan osalla deliverables eli tuotokset. Siinä määritellään, mitä testausprosessin tulisi tuottaa. Testaussuunnitelmassa seuraavana osana on incidentals eli satunnaiset. Tähän sisällytetään kaikki poikkeukset mitä testaukseen liittyy ja kenellä on valtuudet muuttaa testaussuunnitelmaa. Toiseksi viimeisessä osassa on risks eli riskit. Siinä on yksinkertaisesti selitetty testaukseen liittyvät riskit ja miten niitä voisi torjua. Viimeiseen osaan kuuluu tasks eli tehtävät, johon sisältyy testauksen tehtävät. (Kasurinen 2013, 91-92.)

Laajuus on tärkeä osa testaussuunnitelmaa. On tärkeää, että laajuus mietitään kunnolla, jotta ei tulisi testattua turhia tai tarpeettomia osia. Testattavat ja ei-testattavat asia kannataisi listata ja niissä tulisi myös kertoa syyt siihen, miksi tiettyjä kohtia testataan ja tiettyjä kohtia ei testata (Farrel-Vinay 2008, 133). Sovellus Y:ssä tulisi testata kaikki kriittisimmät kohdat, kuten sisäänkirjautuminen, tehtävien teko ja tulosten oikea näkyminen. Kaikkia mahdollisia vääriä syötteitä ei kuitenkaan kannata testata, koska syötteitä on loputon määrä. Sen

sijaan kannattaa miettiä muutamia syötteitä, joilla olisi paras mahdollinen kattavuus testejä tehdessä.

Ihmisten roolien määrittely kannattaa testausuunnitelmassa. Määrittelyjen avulla tiedetään, kuka testausta tekee, kuka testauksen suunnittelee, kuka testausta johtaa ja kuka tarkastaa testien lopputulokset. Aikataulussa tulee olla kokonaisvaltainen testausaikataulu sekä muiden ylimääräisten testien aikataulut. Aikataulu voi sisältää myös arvion siitä, paljon aikaa jokaisen testitapauksen tekemiseen tarvitaan. (Farrell-Vinay 2008, 138-139.) Yritys X käyttää kaikissa projekteissaan scrumia, joten aikataulussa tulee ottaa huomioon sprinttien pituudet. Aikataulu pitäisi miettiä sprinttien mukaan, jotta jokaisen sprintin aikana voitaisiin testata aina uudet ominaisuudet ja kriittiset kohdat.

Lähestymistavassa tulisi määritellä kaikenkattava lähestymistapa testaukseen. Lähestymistavan tulisi varmistaa, että testausta on riittävästi. Sen tulisi sisältää tärkeimmät aktiviteetit, tekniikat ja välineet, joiden avulla testausta suoritetaan. (Farrell-Vinay 2008, 135.)

Testauksen aloitus-, lopetus-, keskeytys- ja jatkamiskriteerien tarkoitus on määritellä, milloin testausta tulisi tehdä ja milloin se tulisi lopettaa. Kannattaa myös määritellä kokonaisvaltainen kriteeri, jolla määritellään onko testaus mennyt läpi vai epäonnistunut. Kriteereissä tulee määritellä tarkasti ehdot, joilla testausta voi jatkaa tai sen voi keskeyttää. Keskeytyskriteereihin usein laitetaan kohdat, joissa kerrotaan monenko tietyn vaikeusasteisen vian jälkeen testaus olisi syytä keskeyttää. (Farrell-Vinay 2008, 135.)

Testausympäristö kohtaan sisältyy tarvittavat sekä halutut testausympäristön ominaisuudet. Ympäristö kohdassa tulisi määritellä käytettävät laitteet, käyttöjärjestelmät sekä muut tarvittavat testausvälineet. Sinne voi myös sisällyttää kaikki muutkin asiat, jota testaamisen tarvitsee, kuten tilat ja ihmiset. (Farrell-Vinay 2008, 138.) Testausta varten tarvitaan eri ympäristöjä, eri laitteita joilla voidaan testata sovellusta. Jos testaus tapahtuu vain yhdellä laitteella, tulokset ovat riittämättömiä, koska ne eivät kata kuin vain yhden ympäristön.

Testausprosessin pitäisi tuottaa iso määrä erilaisia dokumentteja. Näihin dokumentteihin kuuluu esimerkiksi testausuunnitelma, testilogit, vikaraportit, testauksen yhteenvetoraportti sekä dokumentti testausvälineistä. (Farrell-Vinay 2008, 138.) Testausprosessin tulisi tuottaa tuloksia siitä, miten sovellus on vastannut tiettyihin syötteisiin ja tapahtumiin. Tuloksilla itsessään ei ole väliä, vaan sekä hyvät että huonot tulokset ovat yhtä hyviä.

Poikkeavuuksiin kuuluvat kaikki asiat, jotka poikkeavat normaalista testauksesta. Tähän voi kuulua esimerkiksi ihmiset, laitteet tai ohjelmistot, jotka poikkeavat normaalista testauksesta. Se voi myös sisältää tietoja käytetyistä syötteistä tai asioista, jotka kuuluvat mukaan testaukseen vain poikkeustapauksissa.

Riskeihin tulee sisältää tietoja siitä, mitä riskejä testaukseen liittyy. Olisi myös hyvä liittää tietoja epävarmuuksista, kuten mitä tehdään, jos esimerkiksi testaus epäonnistuu ja uusi versio pitäisi saada tuotantoon mahdollisimman pian. (Farrell-Vinay 2008, 140.) Riskeihin usein lisätään myös tieto siitä, miten riskit voitaisiin torjua tai niiden todennäköisyyttä voitaisiin vähentää.

7.2 Testitapausten suunnittelu

Keväällä 2017 kehitettiin ensimmäiset testitapaukset, jotka pohjautuivat ”Ohjelmistotestaus”-kurssin testitapauspohjaan. Silloin ei vielä ollut tarkkaa käsitystä siitä, mitä kohtia testitapauksiin tarvittaisiin tai mitkä kohdat olisivat oleellisia sisällyttää testitapauspohjaan. Tuloksena muodostui testitapauksia, joita alettiin aktiivisesti käyttää yrityksessä. Ajan myötä huomattiin, että testitapaukset olivat turhan monimutkaisia ja niitä oli vaikea käydä läpi nopeasti. Tällöin alettiin miettimään, miten testitapauksia ja itse testausta voitaisiin parantaa.

Uudet testitapaukset suunniteltiin haastattelujen, oman kokemuksen sekä internetistä löydettyjen mallien pohjalta. Tuloksena syntyi kevyemmät ja yksinkertaisemmat versiot testitapauksista ja jonkin verran myös aivan uusia testitapauksia. Uudet testitapaukset ovat paljon nopeampia käydä läpi ja niiden ulkoasu on miellyttävä ja selkeä.

Kaiken aikaa testitapauksia tehdessä on mietitty laadunvarmistusta. Luvussa 3 kerrottiin tarkemmin laadunvarmistuksesta ja laatuattribuuteista. Sovellus Y:n testauksessa on keskitytty testaamaan toiminnallisuutta sekä luotettavuutta. Myös käytettävyys oli osana testitapausten suunnittelua, vaikkakin pienenä osana.

7.3 Testausvälineet

Testausta suunnitellessa tulee miettiä, voiko jo yrityksen olemassaolevia välineitä hyödyntää testauksessa. Tässä tapauksessa yritys X:llä on käytössä Google sheets, joten testitapauksissa hyödynnetään sitä laittamalla testitapaukset sinne. Google sheets on ympäristönä joustava, sillä kaikki oikeudet saaneet henkilöt voivat käydä muokkaamassa ja katsomassa itse testitapauksia. Tätä tutkimusta tehdessä testitapaukset koottiin Excelissä, josta ne myöhemmin siirretään Google Sheetsiin.

Testisuunnitelma sen sijaan on Word-tiedosto, joka tallennettiin Sovellus Y:n omiin tietoihin. Yritys X testaa tälle hetkellä suurimmaksi osaksi kaiken manuaalisesti, vain muutamia testejä on automatisoitu käyttäen postmania. Myöhemmin Yritys X aikoo ottaa käyttöön jonkinlaisen ohjelman, jolla voi testata eri käyttöjärjestelmiä helposti samassa paikassa, kuten esimerkiksi testfairy.com, josta löytyy erilaisia mobiilialustoja.

8 Tutkimuksen tulokset

Tämän tutkimuksen tuloksena syntyi testitapauksia ja testaussuunnitelma Sovellus Y:lle, joka on Yritys X:n kehittämä mobiilisovellus. Vanhoja testitapauksia oli 25 ja uusia testitapauksia on 33. Uusiin testitapauksiin on lisätty mobiilisovellukselle omia testitapauksia sekä mustalaa-tikko-testejä. Testitapauksien kehittämisessä on otettu huomioon haastatteluissa saatu palaute, sekä oma kokemus liittyen testitapauksiin.

Testaussuunnitelma on dokumentti, johon on koottu kaikki Sovellus Y:n mobiilitestaukseen tarvittavat tiedot ja aikataulu. Se sisältää tarkkaa tietoa testauksen kulusta, siihen liittyvistä ihmisistä, välineistä ja riskeistä. Dokumenttia pitäisi päivittää säännöllisesti ja sitä tulisi käyttää ohjenuorana mobiilisovellus Y:n testauksessa. Se on ennen kaikkea hyvä uudelle testaa-jalle tai vain muistin virkistämiseksi.

8.1 Testaussuunnitelma

Testaussuunnitelma päätettiin tehdä käyttämällä SPACE DIRT -menetelmää, josta kerroin lu-vussa 6.2 Testaussuunnitelman laatiminen. Testaussuunnitelma tehtiin yhdessä sovellus Y:n projektipäällikön kanssa. Yritys X aikoo hyödyntää tehtyä testaussuunnitelmaa sovellus Y:n testauksessa ja mahdollisesti myös kehittää sitä vieläkin paremmaksi ja kattavammaksi. Tes-taussuunnitelmaan saatiin tietoja Yritys X:n omista tiedostoista sekä projektin kanssa työsken-televiltä henkilöiltä. Yritys X:llä on ollut joitakin tiedostoja, joissa lukee satunnaisesti tietoja testauksesta, mutta nyt ne on koottu yhteen tiedostoon, testaussuunnitelmaan.

Tätä testaussuunnitelmaa tehdessä on käytetty SPACE DIRT-menetelmää. SPACE DIRT tulee sanoista Scope, People, Approach, Criteria, Environment, Deliverables, Incidentals, Risks ja Tasks. Tämän testaussuunnitelman on tarkoitus antaa suuntaviivoja siihen, miten testausta tulisi suorittaa, jotta se hyödyttäisi itse sovellusta. Testaussuunnitelma sisältää kaikki tarvit-tavat tiedot testauksesta ja sen suorittamisesta kunnolla. Testaussuunnitelman tarkoituksena on auttaa Yritys X:ää testaamaan asioita johdonmukaisesti ja tietyllä kaavalla, jotta testaus olisi mahdollisimman kattavaa.

Sovellus Y:n testaussuunnitelma koskee nimenomaan mobiilisovellus Y:n käyttäjäpuolta. Reg-ressiotestauksessa testataan aina kaikki kriittiset kohdat, jotka on määritelty testitapauksissa tarkemmin. Näitä kohtia ovat muun muassa sisäänkirjautuminen ja tehtävien suorittaminen. Regressiotestaus tulisi suorittaa aina uuden ominaisuuden tai päivityksen myötä, jotta virheet saadaan korjattua mahdollisimman nopeasti. Yritys X käyttää scrumia yrityksen projekteissa ja sprintit kestävät aina yhden viikon. Testausta olisi hyvä tehdä kerran viikossa, mutta kerran kahdessa viikossa vähintään.

Yritys X:llä yksi tietty henkilö hoitaa testauksen. Tämä henkilö testaa kaikki testitapaukset sovittuina aikoina. Suoritettavat testitapaukset valitaan aina edellisviikon perjantaina. Aikataulu on seuraavanlainen: Automaattiset testit suoritetaan maanantaisin ennen klo 10, muut testitapaukset suoritetaan maanantaisin ennen klo 14 ja kuormitustesti suoritetaan maanantaisin testitapauksien jälkeen. Testausta suoritetaan myös tarpeen vaatiessa useammin. Sovelluksen pääkehittäjä päättää testauksesta sekä aikataulusta ja voi halutessaan muuttaa niitä. Jokaiselle testitapaukselle pitäisi varata tietty aika ja tässä tapauksessa jokaisen testitapauksen suorittamiseen on aikaa kolme minuuttia.

Testausmenetelmänä käytetään suurimmaksi osaksi järjestelmätestausta ja osittain mustalaa-tikko-testausta. Regressiotestaus on testauksessa mukana kaiken aikaa, varsinkin uusien päivityksien jälkeen. Testausmenetelmänä kaiken aikaa toimii myös laadunvarmistus, jonka tarkoituksena on pitää huolta siitä, että sovelluksen laatuvaatimukset täyttyvät testattaessa.

Kriteereitä on neljää erilaista, aloitus-, lopetus-, keskeytys- ja jatkamiskriteerejä. Testaus tulisi aloittaa aina sen jälkeen, kun sovellukseen on tullut uusi päivitys tai ominaisuuksia. Toinen syy aloittaa testaus on viikottainen, toistuva testaus. Jotta testaus voidaan aloittaa, kaikki suoritettavat testitapaukset tulee olla määriteltynä. Kaikille Sovellus Y:n vaatimuksille täytyy olla vähintään yksi testi, jotta vaatimukset pystytään testaamaan. Testauksen aloituksen kannalta olennaista on myös se, että kaikki tarvittavat testausvälineet sekä testausympäristöt ovat käytettävissä ja saatavilla.

Joskus testaus täytyy keskeyttää ja tähän syitä voi olla monia erilaisia. Useimmiten testaus joudutaan keskeyttämään, kun on löydetty bugi tai muu virhe, jonka takia testausta ei voida edes suorittaa loppuun asti. Kyseessä on siis sen verran vakava bugi, että se täytyy ensin korjata. Yhtenä syynä testauksen keskeytykseen voidaan pitää vaatimuksien muuttumista. Kesken testauksen sovelluksen vaatimukset ovatkin muuttuneet, jolloin testaus pitää keskeyttää ja testitapauksia pitää muokata tai luoda kokonaan uusia. Testaus voidaan myös keskeyttää, jos yli 40% halutuista toiminnallisuuksista ei toimi.

Jatkamiskriteerit on pääteltävissä keskeytyskriteereistä. Heti kun kriittiset ja melko kriittiset bugit on korjattu, testausta voidaan jatkaa. Vaatimusten muuttuessa tehdään uusi vaatimusmäärittely, jonka jälkeen keskeytetty testaus voidaan aloittaa. Testausta voidaan myös jatkaa, jos uudet testitapaukset on luotu tai vanhoja testitapauksia on muutettu vastaamaan nykyisiä vaatimuksia. Viime kädessä testausta voidaan jatkaa vasta, kun uusin, korjattu versio sovelluksesta on mahdollista testata.

Lopetuskriteereihin sisältyy syitä, minkä takia testaus on syytä lopettaa. Testauksen voi lopettaa, jos testaus on saatu suoritettua loppuun asti ilman ongelmia. Tähän kuuluu se, että kriittisiä tai melko kriittisiä bugeja ei ole löydetty. Testauksen voi myös lopettaa silloin, kun

korjatut bugit on testattu huolella ja mahdolliset löydetty pienemmät bugit ja virheet on kirjattu JIRA:an.

Testausympäristöinä Sovellus Y:n testauksessa toimivat mobiiliympäristöt, koska kyseessä on mobiilisovellus. Testaus tulisi aina suorittaa Android, IOS ja Windows käyttöjärjestelmillä. Käyttöjärjestelmäversion täytyy olla uusin ja puhelinmerkkeinä käytetään Huaweiin, Applen ja Nokian puhelimia. Tarvittaessa testausta voidaan myös suorittaa myös Android, IOS ja Windows tableteilla.

Tuloksena testauksesta pitäisi tulla tietoa siitä, miten sovellus toimii, kuinka hyvin ja miten sovellus toimii varsinkin mobiilissa. Testauksesta kuuluu tulla myös tuloksena kirjallisia dokumentteja, kuten testaussuunnitelma, testilogit, vikaraportteja, testauksen yhteenvedo sekä raportti testausvälineistä. Testilogeihin merkitään mitä testattiin ja miten testit onnistuivat. Jos testauksen aikana vastaan tulee bugeja ja virheitä, tulisi niistä koostaa vikaraportteja JIRA:an. JIRA:sta sovelluksen kehittäjät näkevät, mitä virheitä on tullut vastaan ja kuinka vakavia ne ovat. Tätä kautta vikojen korjaaminen lähtee käyntiin nopeasti. Testauksen yhteenvedoon kirjoitetaan tarkemmin löydettyistä bugeista ja testin eri vaiheista. Testausväline raporttiin kirjataan ylös mitä välineitä testauksessa tarvittiin, mitä ympäristöjä ja mitä tiloja käytettiin. Kaikki tuotokset kirjataan ylös ja ne tallennetaan Google Driveen, josta tulee laittaa linkki Yritys X:n Slackkiin. Virheraportit laitetaan suoraan JIRA:aan.

Poikkeavuutena tässä testaussuunnitelmassa on se, että vain yksi henkilö hoitaa testausta ja testaus tehdään tietyllä käyttäjätunnuksella. Tämä tunnus on tehty nimenomaan testaamista varten. Toisena huomiona voitaisiin sanoa, että kyseessä on nimenomaan mobiilisovellus, joten testaamisessa tulee ottaa huomioon mobiiliympäristö.

Riskinä on, että uudet ominaisuudet tai päivitykset rikkovat jo toimivan mobiilisovelluksen. Tämä on yksi niistä syistä, minkä takia testausta pitää tehdä säännöllisesti. Tähän tulee varautua testaamalla koko sovellus mahdollisimman pian uusimman päivityksen tai ominaisuuden lisäämisen jälkeen. Jos testaus epäonnistuu, voi uuden version julkaisu siirtyä myöhemmäksi. Tämä voi johtaa edelleen isompiin ongelmiin aikataulun ja julkaisun kanssa. Siksi on tärkeää, että testaus suunnitellaan hyvin etukäteen, jotta riskit voidaan minimoida. Riskejä voidaan muutenkin torjua hyvin määrittelemällä laatuvaatimukset ja muut vaatimukset tarkasti. On myös suuri riski hyönteismyrkkyparadoksiin, joten testisuunnitelmia pitäisi uusia säännöllisin väliajoin. Sovellus on laajasti käytetty, joten sen toimimattomuus on erittäin haitallista.

Testitapaukset ovat erillisenä liitteenä. Testitapauksia on laidasta laitaan sisältäen mobiilille erityisiä testitapauksia, järjestelmätestauksen testitapauksia ja musta laatikko -testitapauksia. Osa testitapauksista on tarkoitettu myös regressiotestaukseen.

8.2 Testitapaukset

Testitapaukset päätettiin tehdä käyttäen Google Sheets -ohjelmaa, jossa tehtiin testitapauksille oma taulukko. Taulukko sisältää tietoja siitä, miten testi suoritetaan, millä laitteilla, missä ympäristössä ja kuinka järjestelmän tähän tulisi reagoida. Mukana on 33 kappaletta testitapauksia, jotka kaikki ovat mobiilisovellukselle Y. Testitapauksia suunnitellessa on myös keskitytty siihen, että ne testaavat myös sovelluksen laatua.

Yritys X on muutamien kuukausien aikana käyttänyt minun tekemiä testitapauksia, jotka osoittautuivat liian kömpelöiksi testattavaksi joka viikko. Testitapaukset olivat kaikki omilla välilehdillään ja niiden läpi selaaminen vei aikaa runsaasti. Jokainen testitapaus piti myös täyttää erikseen jokaiselle eri alustalle. Aikaisemmissa tapauksissa oli turhia kohtia, joita ei koskaan käytetty. Taulukko 1:stä voidaan nähdä, että linkki kohta oli hyödytön, sille sovelluksesta oikean paikan löytäminen on helppoa ja linkkejä ei koskaan käytetty. OK? kohta jäi myös käyttämättömäksi, sillä jokaisen testitapauksen tuloksen merkkäminen erikseen osoitautui erittäin hitaaksi, joten Yritys X kehitti itselleen QA Matriisin, josta sovelluksen kehittäjä näkee nopeasti, mitkä testitapaukset ovat onnistuneet ja mitkä ei. Näin ollen myös lopputulos kohta on turha. Muihin tietoihin ei ole koskaan kirjoitettu mitään, joten tämänkin kohdan poisto tuntui luontevalta. Testauspäivämäärä sijoitettiin dokumentin nimeen, joten sekin kohta oli parempi poistaa ja testauksen hoitaa aina sama henkilö, joten Testaaja kohtaa ei erikseen tarvitse.

Testitapauksen sisältö						
Tarkasta omat kommentit						
Toiminto	Oletettu tulos	Linkki (Mistä asian voi tarkistaa)	OK?	Muut tiedot		Testausympäristö
1. Paina "Chat" sivun oikeasta laidasta	Chat sivu aukeaa					(selain, käyttöjärjestelmä, laite yms)
2. Valitse "Omat" sivun ylälaidasta	Uusi sivu avautuu					
3. Tarkasta, että vain sun omat kommentit näkyvät	Vain itsekirjoitetut kommentit näkyvät					
Lopputulos						
Testauspäivä	Testaaja					

Taulukko 1: Vanha testitapaus

Uudet testitapaukset on sijoitettu yhdelle ja samalla sivulle, jotta niiden läpi selaaminen on helpompaa. Vanhoja testitapauksia on muokattu ja uusia testitapauksia on myös kehitetty. Kaikkia testejä kehitettäessä on mietitty, mitä laatuvaatimuksia asiakkaalla on ja mitkä ominaisuudet pitäisi aina testata. Asiakkaalle tärkeintä oli se, että sovellus toimii ja kaikki tehtävät voidaan suorittaa. Myös tulosten tulisi näkyä oikein ja päivittyä samantien, kun muutoksia

tehdään. Samalle sivulle voidaan myös merkitä kaikkien eri alustojen vastaukset, jotta kokonaiskuvan muodostaminen olisi helpompaa ja myös testitapauksien rakennetta on muutettu. Uusissa testitapauksissa on vain kaikki tarpeelliset kohdat, ei mitään ylimääräisiä. Testitapaukset alkavat testauksen kohteella, johon kirjoitetaan mitä ominaisuutta tai asiaa testataan. Seuraavaksi on tavoite, johon merkitään mikä tulisi olla testauksen lopputulos. Tämän jälkeen on toteutus, jossa kerrotaan vaihe vaiheelta, mitä testaajan tulee tehdä, jotta päästään haluttuun lopputulokseen. Seuraavana on lisätiedot -kohta, johon merkitään syyt, miksi testi on mennyt pieleen tai muita huomioita. Siitä seuraavat kolme kohtaa ovat eri mobiilialustat, joihin merkitään millä alustoilla testit menivät läpi ja millä ei.

Taulukosta 2 löytyy muutama testitapaus ja muut testitapaukset löytyvät liitteistä. Mallina yksi mobiilitestitapaus, kriittinen testitapaus ja musta laatikko -testitapaus, johon on laitettu esimerkkivastaukset. Jos kohtiin Android, IOS tai Windows kohtiin kirjoitetaan ok tai ei, niin väri muuttuu ok:n kohdalla vihreäksi ja ei:n kohdalla punaiseksi automaattisesti. Sitten kun sovelluksen kehittäjä käy läpi testitapauksia nopeasti, hän pystyy värien avulla paikantamaan pieleen menneet testitapaukset nopeasti. Lisätiedot kohdassa kerrotaan, mikä on mennyt pieleen tai muita huomiota kyseisestä testitapauksesta.

Testauksen kohde	Tavoite	Toteutus	Lisätiedot	Android 7.0	IOS 11	Windows 10
Akun keston seuranta	Akku ei kulu liikaa testejä tehdessä	1. Katso akun määrä ennen testien aloittamista 2. Katso akun määrä testien loputtua				
Kirjautuminen	Kirjautuminen onnistuu ja oma sivu aukeaa	1. Avaa sovellus Y 2. Kirjoita käyttäjätunnus 3. Kirjoita salasana 4. Paina "kirjaudu" -nappulaa				
Kirjautuminen oikealla käyttäjätunnuksella ja väärällä salasanalla	Kirjautumisnappia voi painaa, mutta tulee teksti "Väärä käyttäjätunnus tai salasana" ja kirjautuminen ei onnistu	1. Avaa sovellus Y 2. Kirjoita oikea käyttäjätunnus 3. Kirjoita väärä salasana 4. Kokeile kirjautua	Kirjautuminen onnistuu väärällä salasanalla	ok	ei	ok

Taulukko 2: Uusia testitapauksia

9 Yhteenveto ja johtopäätökset

Opinnäytetyön aihe saatiin jo keväällä 2017. Aluksi tutustuttiin Sovellus Y:hyn ja sen toimintaan. Sovelluksen kanssa työskentelevä henkilö perehdytti sovelluksen toimintaan, jonka jälkeen sovellukseen pääsi tutustumaan. Tutustumisen kautta alkoi muodostua kuva siitä, miten Sovellus Y toimii ja mitkä toiminnot vaikuttavat mihinkin paikkoihin. Tämän pohjalta aloitettiin tekemään ensimmäisiä testitapauksia ja ne ovatkin olleet Yritys X:llä käytössä siitä asti. Pohja testitapauksiin saatiin koulussa käydyn kurssin ”Ohjelmistotestaus” materiaaleista.

Opinnäytetyön varsinainen kirjoittaminen alkoi lokakuun alussa perehtymällä lähdekirjallisuuteen ja internetissä olevaan materiaaliin. Niiden pohjalta kirjoitettiin teoriaa testauksesta, ketterästä kehityksestä ja laadunvarmistuksesta. Yritys X:ltä saatiin palautetta ja materiaalia

liittyen jo käytössä olleisiin testitapauksiin. Testitapaukset olivat pysyneet samoina muutamia muutoksia lukuunottamatta. Yritys X:ssä toimiva testaaaja kertoi, että testitapaukset olivat hankalia täyttää ja ylimääräisiä kohtia oli jonkin verran. Testitapauksia oli hankala selata ja kaikkien tapausten läpikäyminen yhä uudestaan alkoi käymään raskaaksi. Testaussuunnitelmaa ei ole ollut ollenkaan, vaan yksittäisiä tietoja testaukseen liittyen on kirjoitettu muutama eri paikkoihin.

Uudet testitapaukset ovat kaikki samalla sivulla ja huomattavasti helpompia käyttää. Enää ei tarvitse käydä läpi monia eri välilehtiä, vaan testauksen voi hoitaa sivua kääntämättä. Mukana on myös täysin uusia testitapauksia, jotka kohdistuvat mobiilisovelluksen testaukseen. Kun testitapauksien kohdalle kirjoittaa sanan ”ok”, muuttuu taustaväri vihreäksi ja sanan ”ei” kohdalla väri muuttuu punaiseksi. Näin testitapausten tarkistajana toimiva Sovellus Y:n kehittäjä näkee nopeasti missä kohdissa on tullut virheitä ja millä alustalla. Vanhoissa testitapauksissa ei ollut yhtään testiä liittyen itse mobiilisovellukseen, joten sellaisia lisättiin. Näin itse mobiilisovelluksen käyttöä voidaan myös testata.

Testaussuunnitelmaan on koottu kaikki erillään olleet tiedot mobiilisovellus Y:n testauksesta ja se sisältää myös uudet testitapaukset. Testaussuunnitelman pohjaan saatiin apua kirjallisuudesta. Tietoja testaussuunnitelmaan saatiin haastatteluista, omasta kokemuksesta sekä yrityksen sovellus Y:yn liittyvistä teknisistä dokumenteista.

10 Jatkokehitysehdotukset ja oman oppimisen arviointi

Tulevaisuudessa Yritys X:n kannattaa ajan säästämiseksi automatisoida enemmän testejä. Suurin osa testitapauksista on kriittisiä ja ne täytyy testata joka kerta, kun uusia ominaisuuksia tai muita päivityksiä tulee. Testitapauksiin kannattaa myös mobiilitestien rinnalle lisätä omia testitapauksia tietokoneselainten versioille. Testitapauksia kannattaa myös tehdä sitä mukaa lisää, kun tulee uusia ominaisuuksia tai asioita. Yritys X:n kannattaisi myös aktiivisesti yrittää kehittää lisää testitapauksia, jotta kaikki viat saadaan varmasti havaittua ja korjattua. Uusien testitapauksen kehittäminen myös auttaa siihen, että hyönteismyrkkyparadoksia ei pääse syntymään. Tärkeää testauksen kannalta on se, että testausta laajennetaan ja uusia tapoja otetaan käyttöön. Yritys X:n kannattaisi ehdottomasti harkita harmaalaatikkotestauksen ja lasilaatikkotestauksen mukaanottamista. Käytettävyydestestauksella voitaisiin saada tärkeää tietoa käyttäjäkokemuksesta ja siitä, kuinka sitä voitaisiin parantaa. Yritys X onkin jo jokin verran aloittanut kuormitustestausta ja sitä kannattaa myös jatkossa tehdä säännöllisesti. Yritys X:n kannattaisi myös tulevaisuudessa yrittää testata mobiilialustoja käyttäen esimerkiksi testfairy.com sivustoa, jossa eri alustoja voi testata yhdellä ja samalla sivustolla.

Testaussuunitelmaa Yritys X:n kannattaisi päivittää ja täyttää lisää aina kun tulee uutta asiaan tietoon liittyen sovellus Y:hyn. Testaussuunnitelma kannattaisi myös tehdä Yritys X:n kaikille eri sovelluksille, jotta niitäkin pääsisi johdonmukaisesti testaamaan. Testaaminen on tärkeää, jotta laatuvaatimukset voidaan täyttää.

Opinnäytetyön tekeminen oli kaiken kaikkiaan mukavaa ja mielenkiintoista. Testitapausten miettiminen ja rakentaminen oli haastavaa, mutta palkitsevaa. Opinnäytetyön kirjoittamisen olisi voinut aloittaa aikaisemmin ja koko prosessin olisi voinut miettiä tarkemmin etukäteen. Asioiden kunnollinen aikatauluttaminen olisi varmasti helpottanut prosessia huomattavasti. Prosessi olisi myös ollut vaivattomampi, jos lähdekirjallisuutta olisi etsinyt etukäteen riittävästi. Kirjoittamisen ja uusien testitapausten tekemisen kanssa tuli kiire, jonka takia lopputulos ei ole paras mahdollinen. Testitapaukset ja testaussuunnitelma kuitenkin onnistuivat ja niistä tulee olemaan hyötyä myös tulevaisuudessa. Testitapaukset ja testaussuunnitelma menevät käyttöön Yritys X:llä ja tulevaisuudessa he laajentavat niiden käyttöä myös muihin projekteihin. Uusia testitapauksia tehdessä otettiin esille monia erilaisia esimerkkejä millaisia testitapauksia muualla käytetään ja niiden pohjalta luotiin Yritys X:llä oma testitapauspohja, jonka avulla testauksesta tulisi nopeampaa ja tehokkaampaa, ilman turhia sivujen selauksia. Ideoita testitapausten rakenteisiin saatiin muun muassa koulussa käydystä Ohjelmistotestaus-kurssista.

Lähteet

Painetut

Alasuutari, P. 2011. Laadullinen tutkimus 2.0. 4. Painos. Tampere: Vastapaino

Chemuturi, M. 2011. Mastering Software Quality Assurance. USA: Ross Publishing

Farrell-Vinay, P. 2008. Manage Software Testing. USA: Auerbach Publications

Haikala, I., Märijärvi, J. 2004. Ohjelmistotuotanto. 10. Painos. Hämeenlinna: Karisto

Kananen, J. 2017. Laadullinen tutkimus pro graduna ja opinnäytetyönä. Suomen Yliopistopaino

Kasurinen, J. P. 2013. Ohjelmistotestauksen käsikirja. Jyväskylä: Docendo

Sims, C., Johnson, H.C. 2011. The Elements of Scrum. USA: Dymaxion, LLC.

Tuomi, J., Sarajärvi, A. 2009. Laadullinen tutkimus ja sisällönanalyysi. Jyväskylä: Gummerus Kirjapaino

Vilkka, H., Airaksinen, T. 2004. Toiminnallinen opinnäytetyö. Jyväskylä: Gummerus Kirjapaino

Sähköiset

Chavhan, A. 2015. 7 Fundamental principles of software testing as per ISTQB. Viitattu 25.11.2017

<http://www.softwaretestingandistqb.com/7-fundamental-principles-of-software-testing-as-per-istqb/>

Heinonen, Jarmo. 2012. Luotettavuus. Viitattu 23.11.2017

<https://kyvyt.fi/view/artefact.php?artefact=304009&view=72174>

International Software Test Institute. 2017. What is Software Quality Assurance? Viitattu 29.10.2017

http://www.test-institute.org/What_is_Software_Quality_Assurance.php

ISTQB Exam Certification. 2017. What are the principles of testing? Viitattu 3.11.2017

<http://istqbexamcertification.com/what-are-the-principles-of-testing/>

Lumme, R., Leinonen, R., Leino, M., Falenius, M. & Sundqvist, L. 2006. Monimuotoinen / toiminnallinen opinnäytetyö. Viitattu 3.11.2017

<http://www2.amk.fi/digma.fi/www.amk.fi/opintojak-sot/030906/1113558655385/1154602577913/1154670359399/1154756862024.html>

Tuovinen, A. 2013. Testaus osana ohjelmistojen elinkaarta II. Viitattu 15.10.2017

https://www.cs.helsinki.fi/u/aptuovin/testaus/Ohj_testaus_2013_4.pdf

Ratnakar, N. 2017. Beginner's Guide to Mobile Application Testing. Viitattu 31.10.2017

<http://www.softwaretestinghelp.com/beginners-guide-to-mobile-application-testing/>

Virtuaaliammattikorkeakoulu. 2007. Tutkimuksen validiteetti. Viitattu 23.11.2017

<http://www2.amk.fi/digma.fi/www.amk.fi/opintojak-sot/0709019/1193463890749/1193464185783/1194413809750/1194415367669.html>

Julkaisemattomat

Henkilö 1. 2017. Testaajan haastattelu 29.10.2017. Yritys X. Helsinki

Henkilö 2. 2017. Sovellus Y:n kehittäjän haastattelu 26.10.2017. Yritys X. Helsinki

Kuviot

Kuvio 1: Laatuattribuuttien lähestymistapa (International Software Test Institute 2017) 10

Kuvio 2: Mustalaatikko -testaus (Kasurinen 2013, 52) 12

Taulukot

Taulukko 1: Vanha testitapaus 22

Taulukko 2: Uudet testitapaukset 23

Liitteet

Liite 1: Testitapaukset	30
Liite 2: Testaussuunnitelma.....	33

Liite 1: Testitapaukset

Mobiilitestit						
Testauksen kohde	Tavoite	Toteutus	Lisätiedot	Android 7.0	IOS 11	Windows 10
Akun keston seuranta	Akku ei kulu liikaa testejä tehdessä	1. Katso akun määrä ennen testien aloittamista 2. Katso akun määrä testien loputtua				
Sovelluksen toiminnallisuus	Sovellus ei kaadu nettiongelmien takia	1. Avaa sovellus 2. Laita netti pois päältä				
Sovelluksen nopeuden seuranta	Kuinka nopeasti sovellus reagoi eri asioihin	1. Tee kaikki testit 2. Kiinnitä testien aikana huomiota sovelluksen reaktiokykyyn				
Asennus ja lataaminen	Sovellus Y latautuu ja asentuu normaalisti	1. Lataa sovellus 2. Asenna sovellus				

Käyttäjä						
Testauksen kohde	Tavoite	Toteutus	Lisätiedot	Android 7.0	IOS 11	Windows 10
Käyttäjätietojen muuttaminen (Puhelinnumero, s-posti, kieli)	Käyttäjätietoja voi muuttaa ja tallentaa	1. Avaa sovellus Y 2. Paina omaa nimeäsi/kuvaasi vasemmasta yläkulmasta 3. Paina "Muokkaa" oikeasta yläkulmasta 4. Paina "Valitse kuva" muuttaaksesi käyttäjäkuvan 5. Muuta s-postiosoite 6. Muuta puhelinnumero 7. Muuta kieli 8. Kirjoita uusi salasana kahdesti 9. Paina "Tallenna" vasemmasta yläkulmasta				
Projektin vaihtaminen	Projektia voi vaihtaa	1. Avaa sovellus 2. Paina omaa nimeäsi/kuvaasi vasemmasta yläkulmasta 3. Paina nykyistä projektin nimeä 4. Vaihda projekti johonkin toiseen				

Perustoiminnot						
Testauksen kohde	Tavoite	Toteutus	Lisätiedot	Android 7.0	IOS 11	Windows 10
Sovellus Y:n avaus	Sovellus Y avautuu	1. Valitse sovellus Y ja avaa se.				
Kirjautuminen	Kirjautuminen onnistuu ja oma sivu aukeaa	1. Avaa sovellus Y 2. Kirjoita käyttäjätunnus 3. Kirjoita salasana 4. Paina "kirjaudu" -nappulaa				
Salasanan resetointi	Uusi salasana saapuu sähköpostiin ja sillä pystyy kirjautumaan sisälle	2. Paina "Unohtuiko salasana" 3. Kirjoita sähköpostiosoite tyhjään kohtaan 4. Paina "Läheta"				
Paluu etusivulle	Sovelluksessa oleva Sovellus Y:n logon pitäisi viedä takaisin etusivulle	1. Avaa sovellus Y 2. Paina "Chatti" sivun oikeasta reunasta 3. Paina sovellus Y:n logoa				

Vastaukset ja omat tehtävät						
Testauksen kohde	Tavoite	Toteutus	Lisätiedot	Android 7.0	IOS 11	Windows 10
Tällä viikolla tehtyjen tehtävien prosenttien tarkistaminen	Kaikki tehdyt tehtävät näkyvät ja niiden vastausprosentteja voi tarkastella	1. Avaa sovellus Y 2. Paina "Historia" sivun yläalaidasta 3. Tarkasta, että tehtyjen tehtävien prosentit ovat oikein				
Viime kuussa tehtyjen tehtävien prosenttien tarkistaminen	Viime kuussa tehdyt tehtävät näkyvät ja niiden vastausprosentteja voi tarkastella	1. Avaa sovellus Y 2. Paina "Historia" sivun yläalaidasta 3. Valitse jokin viikko yli kuukauden takaa 4. Tarkasta, että tehtyjen tehtävien prosentit ovat oikein 5. Tarkasta, että prosenttien alapuolella kommentit näkyvät oikein				
Omien tehtävien katsominen	Omissa tehtävissä näkyvät oikeat tehtävät	1. Avaa sovellus Y 2. Paina "Historia" sivun yläalaidasta 3. Paina "tehtäväni" vasemmasta yläreunasta 4. Paina mitä tahansa tehtävää 5. Tarkasta, että tehtävän tiedot on oikein				

Chatti						
Testauksen kohde	Tavoite	Toteutus	Lisätiedot	Android 7.0	IOS 11	Windows 10
Kommentin ja kuvan lisääminen keskustelualueelle	Kommentin ja kuvan lisääminen onnistuvat	1. Avaa sovellus Y 2. Paina "Chatti" sivun oikeasta laidasta 3. Kirjoita jotain kommenttiriville 4. Paina pientä kuva ikonia kommenttirivin alapuolella 5. Valitse kuva 6. Paina "Lähetä"				
Kommenttiin vastaaminen	Vastaaminen onnistuu	1. Avaa sovellus 2. Paina "Chatti" sivun oikeasta laidasta 3. Valitse kommentti, johon haluat vastata 4. Paina "kommentoi" 5. Kirjoita jotain 6. Paina pientä kuva ikonia kommenttirivin alapuolella 7. Paina "kommentoi"				
Ilmoitusten tarkistus	Ilmoituksia voi tarkastella ja kaikki oikeat ilmoitukset näkyvät	1. Avaa sovellus Y 2. Paina "Chatti" sivun oikeasta laidasta 3. Valitse "Ilmoitukset" sivun vasemmasta yläkulmasta 4. Tarkasta näkykö kaikki oikeat ilmoitukset				
Kommentin muokaus ja poistaminen	Omia kommentteja voi muokata tai poistaa halutessaan	1. Avaa sovellus Y 2. Paina "Chatti" sivun oikeasta laidasta 3. Etsi postaus, jossa on testikäyttäjän kommentti ja paina "[numero] kommentteja" 4. Paina rattaan kuvaa kommenttisi yläpuolelta 5. Paina "Muokkaa" 6. Muokkaa kommenttia ja paina "Peruuta" 7. Toista kohdat 3 ja 4 8. Muokkaa kommenttia ja paina "Tallenna" 9. Paina rattaan kuvaa kommenttisi yläpuolelta 10. Paina "Poista" 11. Paina "Kyllä"				

Tulokset						
Testauksen kohde	Tavoite	Toteutus	Lisätiedot	Android 7.0	IOS 11	Windows 10
Tarkasta tuloksien prosentit/määrät	Tuloksien prosentit ovat muuttuneet tehtävien tekemisien myötä	1. Avaa sovellus Y 2. Paina "Tulokset" sivun yläaidasta 3. Tarkasta, että prosentit näkyvät oikein 4. Valitse jokin toinen viikko 5. Tarkasta, että prosentit näkyvät oikein 6. Paina "vastausmäärät" prosenttien oikealta puolelta 7. Tarkasta, että määrät ovat oikein				
Tarkasta vastaukset tuloksista	Tulokset ovat saatavilla ja ne näkyvät oikein	1. Avaa sovellus Y 2. Paina "Tulokset" sivun yläaidasta 3. Rullaa alas ja tarkista, että vastaukset ovat oikein ja että kaikki näkyvät oikein.				
Tarkasta tulokset Tulokset -sivulta	Tulokset näkyvät oikein ja sivu avautuu normaalisti	1. Avaa sovellus Y 2. Paina "Tulokset" sivun yläaidasta 3. Rullaa alas ja klikkaa "Tulokset" 4. Tarkasta, että tiedot on oikein				
Tarkasta vastauksien prosentit	Sivu avautuu normaalisti ja tulokset näkyvät oikein	1. Avaa sovellus Y 2. Paina "Tulokset" sivun yläaidasta 3. Rullaa alas ja klikkaa "Prosentit" 4. Tarkasta, että tiedot ovat oikein				
Tarkasta tietyn tehtävän tulokset	Sivu avautuu normaalisti ja tiedot ovat oikein	1. Avaa sovellus Y 2. Paina "Tulokset" sivun yläaidasta 3. Rullaa alas ja klikkaa "Prosentit" 4. Valitse jokin tehtävä ja paina pientä nuolen kuvaa, joka osoittaa alaspäin 5. Tarkasta, että tiedot näkyvät oikein				

Kysymyksiin ja tehtäviin vastaaminen						
Testauksen kohde	Tavoite	Toteutus	Lisätiedot	Android 7.0	IOS 11	Windows 10
Vastaaminen kyllä/ei kysymykseen ja tuloksien muutosten seuraaminen	Kysymykseen voi vastata ja vastaus muuttaa tulosnäkyä	1. Avaa sovellus Y 2. Katso tulokset "Historia" -sivulta ja paina ne mieleen 3. Katso tulokset "Tulokset" -sivulta ja paina ne mieleen 4. Valitse kyllä/ei -tehtävä 5. Vastaa kyllä tai ei 6. Katso muuttuivatko tulokset "Historia" -sivulla 7. Katso muuttuivatko tulokset "Tulokset" -sivulla				
Vastaaminen avoimeen kysymykseen ja tuloksien muutosten seuraaminen	Kysymykseen voi vastata ja vastaus muuttaa tulosnäkyä	1. Avaa sovellus Y 2. Katso tulokset "Historia" -sivulta ja paina ne mieleen 3. Katso tulokset "Tulokset" -sivulta ja paina ne mieleen 4. Valitse avoin kysymys -tehtävä 5. Vastaa kysymykseen 6. Katso muuttuivatko tulokset "Historia" -sivulla 7. Katso muuttuivatko tulokset "Tulokset" -sivulla				
Vastaaminen numeeriseen kysymykseen ja vastausten muutosten seuraaminen	Kysymykseen voi vastata ja vastaus muuttaa tulosnäkyä	1. Avaa sovellus Y 2. Katso tulokset "Historia" -sivulta ja paina ne mieleen 3. Katso tulokset "Tulokset" -sivulta ja paina ne mieleen 4. Valitse numeerinen kysymys -tehtävä 5. Vastaa kysymykseen 6. Katso muuttuivatko tulokset "Historia" -sivulla 7. Katso muuttuivatko tulokset "Tulokset" -sivulla				
Vastaaminen äänestys tehtävään ja tuloksien muutosten seuraaminen	Tehtävään voi vastata äänestämällä ja vastaus muuttaa tulosnäkyä	1. Avaa sovellus Y 2. Katso tulokset "Historia" -sivulta ja paina ne mieleen 3. Katso tulokset "Tulokset" -sivulta ja paina ne mieleen 4. Valitse äänestys -tehtävä 5. Valitse jokin vaihtoehtoista 6. Katso muuttuivatko tulokset "Historia" -sivulla 7. Katso muuttuivatko tulokset "Tulokset" -sivulla				
Vastaaminen media tehtävään ja tuloksien muutosten seuraaminen	Tehtävään voi vastata ja vastaus muuttaa tulosnäkyä	1. Avaa sovellus Y 2. Katso tulokset "Historia" -sivulta ja paina ne mieleen 3. Katso tulokset "Tulokset" -sivulta ja paina ne mieleen 4. Valitse media -tehtävä 5. Paina "Valitse Galleriasta" 6. Valitse jokin mediatiedosto ja paina ok 7. Katso muuttuivatko tulokset "Historia" -sivulla 8. Katso muuttuivatko tulokset "Tulokset" -sivulla				
Vastaaminen konsolidoivaan kysymykseen ja tuloksien muutosten seuraaminen	Kysymykseen voi vastata ja vastaus muuttaa tulosnäkyä	1. Avaa sovellus Y 2. Katso tulokset "Historia" -sivulta ja paina ne mieleen 3. Katso tulokset "Tulokset" -sivulta ja paina ne mieleen 4. Valitse konsolidoiva kysymys -tehtävä 5. Valitse jokin kuva 6. Katso muuttuivatko tulokset "Historia" -sivulla 7. Katso muuttuivatko tulokset "Tulokset" -sivulla				

Syötetestit						
Testauksen kohde	Tavoite	Toteutus	Lisätiedot	Android 7.0	IOS 11	Windows 10
Kirjautuminen ilman käyttäjätunnusta	Kirjautumisnappia ei voi painaa ja enterin painaminen ei tee mitään	1. Avaa sovellus Y 2. Kirjoita salasana 3. Jätä käyttäjätunnus kohta tyhjäksi 4. Kokeile kirjautua				
Kirjautuminen ilman salasanaa	Kirjautumisnappia ei voi painaa ja enterin painaminen ei tee mitään	1. Avaa sovellus Y 2. Kirjoita käyttäjätunnus 3. Jätä salasana kohta tyhjäksi 4. Kokeile kirjautua				
Kirjautuminen ilman käyttäjätunnusta ja salasanaa	Kirjautumisnappia ei voi painaa ja enterin painaminen ei tee mitään	1. Avaa sovellus Y 2. Jätä salasana ja käyttäjätunnus kohta tyhjäksi 3. Kokeile kirjautua				
Kirjautuminen oikealla käyttäjätunnuksella ja väärällä salasanalla	Kirjautumisnappia voi painaa, mutta tulee teksti "Väärä käyttäjätunnus tai salasana" ja kirjautuminen ei onnistu	1. Avaa sovellus Y 2. Kirjoita oikea käyttäjätunnus 3. Kirjoita väärä salasana 4. Kokeile kirjautua				
Kirjautuminen oikealla salasanalla ja väärällä käyttäjätunnuksella	Kirjautumisnappia voi painaa, mutta tulee teksti "Väärä käyttäjätunnus tai salasana" ja kirjautuminen ei onnistu	1. Avaa sovellus Y 2. Kirjoita väärä käyttäjätunnus 3. Kirjoita oikea salasana 4. Kokeile kirjautua				

Liite 2: Testaussuunnitelma

Sovellus Y Yritys X Projektipäällikkö Matti Meikäläinen Versio 1	25.11.17
---	----------

Sovellus Y:n testaussuunnitelma

Tätä testaussuunnitelmaa tehdessä on käytetty SPACE DIRT-menetelmää. SPACE DIRT tulee sanoista Scope, People, Approach, Criteria, Environment, Deliverables, Incidentals, Risks ja Tasks. Tämän testaussuunnitelman on tarkoitus antaa suuntaviivoja siihen, miten testausta tulisi suorittaa, jotta se hyödyttäisi itse sovellusta.

Tämä dokumentti sisältää kaikki tarvittavat tiedot testauksesta ja sen suorittamisesta.

Testauksen tarkoituksena on saada välitöntä palautetta sovelluksen kehittäjille seuraavista asioista:

- Ominaisuuksien status
- Julkaisun status
- Julkaisun viivästymisen tai epätunnistumisen mahdollisuudesta
- Sovelluksen suorituskyvystä
- Vaatimusten eroavaisuuksista kehittäjien ja asiakkaan välillä

1. Scope eli Laajuus

Tämä testaussuunnitelma koskee nimenomaan mobiilisovellus Y:n käyttäjäpuolta. Admin puolelle tulisi tehdä oma testaussuunnitelma, koska käyttäjäpuoli ja admin puoli ovat liian suuret, jotta ne voitaisiin yhdistää samaan testaussuunnitelmaan. Regressiotestauksessa testataan aina kaikki kriittiset kohdat, jotka on määritelty testitapauksissa tarkemmin. Näitä kohtia ovat muun muassa sisäänkirjautuminen ja tehtävien suorittaminen. Regressiotestaus tulisi suorittaa aina uuden ominaisuuden tai päivityksen myötä, jotta virheet saadaan korjattua mahdollisimman nopeasti.

Yritys X käyttää scrumia yrityksen projekteissa ja sprintit kestävät aina yhden viikon. Testausta olisi hyvä tehdä kerran viikossa, mutta kerran kahdessa viikossa vähintään. Regressio ja musta laatikko -testaukset suoritetaan aina tarpeen vaatiessa.

2. People eli Ihmiset

Yritys X:llä yksi tietty henkilö hoitaa testauksen. Tämä henkilö testaa kaikki testitapaukset sovittuina aikoina. Suoritettavat testitapaukset valitaan aina edellisviikon perjantaina. Aikataulu on seuraavanlainen:

Automaattiset testit suoritetaan maanantaisin ennen klo 10

Muut testitapaukset suoritetaan maanantaisin ennen klo 14

Kuormitustesti suoritetaan maanantaisin testitapauksien jälkeen

Sovellus Y
Yritys X
Projektipäällikkö Matti Meikäläinen
Versio 1

25.11.17

Testausta suoritetaan myös tarpeen vaatiessa useammin. Sovelluksen pääkehittäjä päättää testauksesta sekä aikataulusta ja voi halutessaan muuttaa niitä. Jokaisen testitapauksen suorittamiseen on varattu kolme minuuttia aikaa.

3. Approach eli Lähestymistapa

Testaustapoina käytetään järjestelmä sekä musta laatikko -testausta yhdistettyinä laadunvarmistukseen. Regressiotestausta tapahtuu aina uusien päivitysten ja ominaisuuksien myötä.

4. Criteria eli Kriteerit

Aloituskriteerit:

- On tullut uusi ominaisuus tai päivitys
- Viikottaisen testauksen aika
- Kaikki suoritettavat testitapaukset on määritelty
- Kaikille vaatimuksille on ainakin yksi testi
- Kaikki testausvälineet on saatavilla ja käytettävissä

Lopetuskriteerit:

- Testaus on suoritettu onnistuneesti loppuun asti.
- Kriittisiä bugeja ei ole löydetty
- Kriittiset bugit on korjattu ja testattu
- Bugit ja muut virheet on kirjattu JIRAAN

Keskeytyskriteerit:

- Kriittinen bugi löydetty
- Melko kriittisiä bugeja löydetty kolme kappaletta
- Vaatimukset ovat muuttuneet
- Yli 40% halutuista toiminnallisuuksista ei toimi

Jatkamiskriteerit:

- Kriittiset ja melko kriittiset bugit on korjattu
- Vaatimusmäärittely on tehty uudelleen
- Testitapauksia on muutettu vastaamaan nykyisiä vaatimuksia
- Uusi korjattu versio on julkaistu

5. Environment eli Ympäristö

Sovellus Y
Yritys X
Projektipäällikkö Matti Meikäläinen
Versio 1

25.11.17

Testaus tulisi aina suorittaa seuraavissa mobiiliympäristöissä:

Android puhelin – Käyttöjärjestelmäversio: uusin – merkki Huawei

iOS puhelin – Käyttöjärjestelmäversio: uusin – merkki Apple

Windows puhelin – Käyttöjärjestelmäversio: uusin – merkki Nokia

Tarvittaessa:

Android tabletti – Käyttöjärjestelmäversio: uusin – merkki Huawei

iOS tabletti – Käyttöjärjestelmäversio: uusin – merkki Apple

Windows tabletti – Käyttöjärjestelmäversio: uusin – merkki Nokia

Testauksessa käytetään laitteita, joita Yritys X:ltä löytyy ja valmiina. Testaus suoritetaan aina Yritys X:n tiloissa.

6. Deliverables eli Tuotokset

Tuloksena testauksesta pitäisi tulla tietoa siitä, miten sovellus toimii, kuinka hyvin ja miten sovellus toimii varsinkin mobiilissa. Testauksesta pitäisi tulla myös kirjallisia tuotoksia, kuten:

- Testaussuunnitelma
- Testilogit
- Vikaraportit (JIRA)
- Testauksen yhteenveto
- Testausväline -raportti

Kaikki tuotokset kirjataan ylös ja ne tallennetaan Google Driveen, josta tulee laittaa linkki Yritys X:n Slackiin.

7. Incidentals eli Satunnaisuudet

Sovellus Y
Yritys X
Projektipäällikkö Matti Meikäläinen
Versio 1

25.11.17

Testaus suoritetaan tietyillä käyttäjätunnuksilla. Testauksen kohteena on vain mobiilisovellus. Ainut henkilö, joka saa muuttaa testaussuunnitelmaa on projektipäällikkö.

8. Risk eli riskit

Riskinä on, että uudet ominaisuudet tai päivitykset rikkovat jo toimivan mobiilisovelluksen. Tämä on yksi niistä syistä, minkä takia testausta pitää tehdä säännöllisesti. Jos testaus epäonnistuu, voi uuden version julkaisu siirtyä myöhemmäksi, koska havaitut virheet tulee korjata ja kaikki testata uudelleen. Riskejä voidaan torjua hyvin sillä, että laatuvaatimukset ja muut vaatimukset ovat tarkkaan määritelty ennakoin testausta aloitetaan. Myös hyvä testauksen suunnittelu ennen testausta auttaa testauksen sujumisen jouhevammin.

9. Tasks eli Tehtävät

Testitapaukset ovat erillisenä liitteenä. Testitapauksia on laidasta laitaan sisältäen mobiilille erityisiä testitapauksia, järjestelmätestaukse testitapauksia ja musta laatikko -testitapauksia. Osa testitapauksista on tarkoitettu myös regressiotestaukseen.

